Southern African Linguistics and Applied Language Studies 2008, 26(2): 209–216
Printed in South Africa — All rights reserved

SOUTHERN AFRICAN LINGUISTICS AND APPLIED LANGUAGE STUDIES
ISSN 1607–3614 EISSN 1727–9461
DOI: 10.2989/SALALS.2008.26.2.3.567

Containing overgeneration in Zulu computational morphology[1]

Laurette Pretorius[1] and Sonja E Bosch[2]

[1]School of Computing, University of South Africa, PO Box 392, UNISA 0003, Pretoria, South Africa
and
Knowledge Systems Group, Meraka Institute, CSIR, Pretoria, South Africa
e-mail: pretol@unisa.ac.za

[2]Department of African Languages, University of South Africa, PO Box 392, UNISA 0003, Pretoria, South Africa
e-mail: boschse@unisa.ac.za

Abstract: The development of a large-coverage, computational morphological analyser for Zulu requires the modelling not only of the regular phenomena often associated with word formation, but also the idiosyncratic behaviour that may occur in Zulu morphology. This paper discusses the application of an existing rule-based, finite-state morphological analyser prototype **ZulMorph** in semi-automating the mining of available Zulu language corpora for idiosyncratic behaviour. The semi-automated procedure makes provision for bootstrapping the morphological analyser to include newly extracted information from corpora. Of particular interest is also the central role that the machine-readable lexicon plays. The procedure is applied to a Zulu development corpus of 30 000 types and the results are given and discussed.

Introduction

Although the view that the problem of computational morphology has been conceptually solved was expressed a number of years ago (Cole *et al*., 1997: 96), it is still not the case that real large-coverage morphological analysers exist for most languages of the world. The Bantu language most well known for its extensive technological development is Swahili (Hurskainen, 1992; Hurskainen, 2004). Zulu, also a member of the Bantu language family, is one of the official indigenous languages of South Africa for which finite-state computational morphological analysis has been reported on (Pretorius & Bosch, 2003a). In this article we describe how a rule-based, finite-state morphological analyser prototype for Zulu may be extended to a tool that may be considered a real large-coverage Zulu morphological analyser. The present focus is on addressing the problem of overgeneration.

Overgeneration occurs when a morphological analyser based on word formation rules analyses or generates entities that are well formed according to these rules, but are in fact invalid strings or illicit structures in terms of the real language. Since the ultimate aim of computational morphology is to model and implement the prevalent linguistic phenomena accurately and efficiently, any source of inaccuracy or spurious word formation or analysis, such as overgeneration, should ideally be prevented, but certainly contained as far as possible. A precise treatment of this phenomenon remains a challenge not only in computational morphology, but also in NLP in general. This is particularly true when the focus is on large-coverage NLP/HLT tools for real applications, where the emphasis is on both analysis and generation, and where acceptable accuracy is required.

The morphological structure of most natural languages, including Zulu, is sufficiently regular to render finite-state computational morphological analysis a standard and state-of-the-art approach (Beesley & Karttunen, 2003). However, even at the morphological level natural languages often exhibit idiosyncratic morphological behaviour, that is, behaviour that cannot be captured by means of rules and regular expressions. The question as to how this impacts the development of a large-coverage computational morphological analyser for a morphologically complex language such as Zulu therefore needs to be considered.

In order to obtain such idiosyncratic information, a hybrid approach is proposed. Language corpora are used in combination with the rules and the grammar embodied in the Zulu morphological analyser and its underlying lexicon of explicitly enumerated noun stems and verb roots. Idiosyncratic behaviour is identified on the basis of valid rule applications that lead to illicit morphological analyses. Once identified, such idiosyncrasies are stored in the machine-readable (MR) lexicon (to be distinguished from the above-mentioned underlying stem/root lexicon of the morphological analyser). This MR lexicon is based on a Bantu languages data model (Bosch *et al*., 2006) and plays a central role in the sense that it serves as an appropriately structured repository for all the available lexical information, idiosyncratic or otherwise. Current prototypes of such resources are a lexicon with 26 000 entries; a finite-state morphological analyser, including 13 000 noun stems and 7 400 verb roots explicitly enumerated; and a development corpus of 30 000 types.

In this article we consider two instances of idiosyncratic behaviour prevalent in Zulu morphology and propose a procedure for appropriately incorporating them in the finite-state morphological analyser. The remainder of the paper is structured as follows: In the next section we discuss the linguistic principles of Zulu morphology and give examples of sources of overgeneration. In the subsequent section the focus is the general approach to Zulu computational morphology and a semi-automated procedure for containing overgeneration. The penultimate section contains a discussion of examples and results, followed by the conclusion and future work.

Overgeneration in Zulu computational morphology

Linguistic principles of Zulu morphology

The rich, agglutinating morphological structure which characterises a language such as Zulu, is based on two principles, namely, the nominal classification system and the concordial agreement system. According to the nominal classification system, nouns are categorised by prefixal morphemes, which for analysis purposes have been put into classes and given numbers. These noun class prefixes bring about concordial agreement that links the noun to other words in the sentence, such as verbs, adjectives and pronouns.

In this article the focus is on processes of derivation and inflection as potential sources of overgeneration in Zulu. Derivational morphology is a combination of morphemes, which produces a new word in a different word category. Examples are nouns derived from verb roots, necessitating a noun prefix as well as a deverbative suffix; or adverbs derived from nouns, necessitating locative prefixes and sometimes suffixes as well. Inflectional morphology is the inclusion of morphemes in a word that do not change the word category, but add information such as tense, aspect, person and number. In the case of nouns as well as verbs, prefixes and suffixes function as inflectional morphemes.

Sources of overgeneration

Overgeneration in Zulu computational morphology is mainly caused by constructions which are not strictly rule based. Although such constructions may be well formed according to the rules, they are in fact invalid strings or illicit structures in terms of the real language. Examples are locatives derived from nouns, as well as the extension of verb roots by means of suffixes.

In general, Bantu languages do not have any prepositions, and therefore prepositional phrases introduced by ‘to/at/in/from/on’ in English, for instance, are derived from other word categories (Poulos & Msimang, 1998: 395). In the derivational morphology of Zulu, adverbs denoting location may be derived from nouns by prefixing a locative prefix *ku-* in the case of nouns in classes 1, 1a, 2 and 2a; or a locative prefix *e-* followed by a locative suffix *-ini* in noun classes 3 to 10; and in exceptional cases by prefixation of the prefix *e-* only in classes 3 to 10, for example:

1(a) *ubaba* ‘father’
ku-u-baba > *kubaba* ‘to/at father’
loc.pref-class1a.noun.pref-noun.stem

1(b) *intaba* ‘mountain’
e-in-ntaba-ini > *entabeni* ‘on the mountain’
loc.pref-class9.noun.pref-noun.stem-loc.suf

1(c) *ikhaya* 'home'
e-i(li)-khaya > *ekhaya* 'at home'
loc.pref-class5.noun.pref-noun stem

Whereas the locative formation of nouns in classes 1, 1a, 2 and 2a, as illustrated in example 1(a), is predictable, the other two methods of deriving locatives are not rule based, and therefore not predictable. Furthermore, the MR lexicon under development for Zulu does not at this stage contain information on the latter type of locative derivatives.

In the inflectional morphology of Zulu, the basic meaning of a verb root may be modified by suffixing so-called extensions to the verb root, functioning as inflectional morphemes, for example:

2(a) *-bon-a* > *-bona* 'see'
-verb.root-terminative

2(b) *-bon-is-a* > *-bonisa* 'show'
-verb.root-caus.ext-terminative

2(c) *-bon-an-a* > *-bonana* 'see each other'
-verb.root-reciproc.ext-terminative

2(d) *-bon-is-an-a* > *-bonisana* 'show each other'
-verb.root-caus.ext-reciproc.ext-terminative

As illustrated in 2(b-d), verbal extension suffixes modify the basic meaning of the verb root. In certain cases, as illustrated in 2(d), more than one extension may even be added.

However, not all roots may take all extensions arbitrarily owing to restrictions on the combinations of certain meanings (Poulos & Msimang, 1998: 183). The following example is ungrammatical because the neuter extension is incompatible with the meaning of the verb 'thunder' and therefore represents a semantic restriction:

3 **-dum-ek-a* > *-dumeka* '*thunderable(?)'
-verb.root-neut.ext-terminative

Regarding sequences of combinations, the passive is used as example: In a sequence of two or more extensions, the passive is usually last in the sequence. However, in the case of certain verb roots, the reciprocal extension follows the passive, while both sequences are possible in other cases, for example:

4(a) *-bon-an-w-a* > *-bonanwa* 'seen by each other'
-verb.root-recip.ext-pass.ext-terminative

4(b) *-bon-w-an-a* > *-bonwana* 'seen by each other'
-verb.root-pass.ext-recip.ext-terminative

There are also instances where the applied extension follows the passive (cf. Van Eeden, 1956: 657),

5 *ya-bulal-w-el-a* > *yabulawela* 'he was killed for'
subj.conc-verb.root-pass.ext-appl.ext-terminative

The previously mentioned MR lexicon does also not yet contain exhaustive information on the combinations and sequences of extensions with verb roots. This type of information could, however, be extracted from language corpus resources, preferably by semi-automated procedures.

ZulMorph, the finite-state computational morphological analyser for Zulu under discussion, is inherently rule based and consequently prone to overgeneration. The challenge is therefore to retain and exploit the modelling and implementation elegance and efficiency of the finite-state paradigm while also providing linguistic and computational mechanisms to capture non-rule-based linguistic information from, for instance, language corpus resources in order to contain phenomena such as overgeneration, and to include such information in the MR lexicon.

Containing overgeneration

General computational approach

The Xerox finite-state tools (Beesley & Karttunen, 2003) are well known as one of the preferred toolkits for modelling and implementing natural language morphology. Its **lex**icon **c**ompiler, **lexc**, is well suited to capturing the morphotactics of Zulu. A **lexc** script, consisting of cascades of

so-called continuation classes (of morpheme lexicons) representing the (concatenative) morpheme sequencing, is compiled into a finite-state network. The Xerox regular expression language, **xfst**, provides an extended regular expression calculus with sophisticated Replace Rules for describing the morphophonological alternation rules of Zulu. The **xfst** script is also compiled into a finite-state network. These networks are finally combined by means of the operation of composition into a so-called Lexical Transducer that constitutes the morphological analyser and contains all the morphological information of Zulu, including derivation, inflection, alternation and compounding (Pretorius & Bosch, 2003b).

The Xerox toolkit also offers the notion of *flag diacritics* as an elegant and efficient means of feature-setting and feature-unification. They may be used to block illegal paths at run time by the analysis and generation routines. In **lexc** and **xfst** they are treated as multicharacter symbols spelt according to these templates: `@operator.feature@` and `@operator.feature.value@`.

Typical operators are Unification, Positive Setting, Require Test, and Disallow Test (Beesley & Karttunen, 2003). Flag diacritics in a network are not matched against the input, and they are not produced as output. Rather, they are feature-based actions that are recognised and performed by the application (analysis or generation) code. When, during an analysis, a root marked `@P.suffX.ON@` is matched, the `@P.suffX.ON@` action causes the feature `suffX` to be set 'Positively' to `ON` in a small feature table maintained by the analysis code. If the suffix itself has a flag diacritic `@R.suffX.ON@` as part of its spelling, then any analysis through the suffix path 'Requires' that the feature table contain the value `suffX=ON` at the time that the suffix is matched. This and similar techniques can be used to handle inter-morpheme dependencies within words, even separated dependencies. In subsequent subsections it will be used to mark particular idiosyncratic behaviour.

The marking of roots with P-type flags, correlated with suffixes marked with R-type flags, is appropriate when the roots marked with P-type flags represent exceptions to the rule. There are also typically cases in natural language where a particular affix (say a suffix) is highly productive and is taken by most roots, with relatively few exceptions. In such cases, it is customary to let *unmarked roots* take the suffix and *mark only the exceptional roots* with flag diacritics, for example, mark only the exceptional root with `@P.suffY.OFF@` or `@U.SuffY.OFF@` and mark the usually productive suffix as `@U.suffY.ON@` (see *-banga* versus *-khaya* in Figure 1).

Semi-automated procedure

Marking locative information

Ideally the ultimate morphological analyser should contain an enumeration of all the known noun stems in Zulu, all marked with appropriate locative information.

```
! Multicharacter Symbols
@P.LocSuf.OFF@ @R.LocSuf.OFF@ @U.LocSuf.ON@

LEXICON NStem
...
banga                NClass5-6; !the rule
khaya@P.LocSuf.OFF@  NClass5-6; !the exception
...

LEXICON NominalSuffixes
...
ini[LocSuf]@U.LocSuf.ON@ : ini@@U.LocSuf.ON@ #;
@R.LocSuf.OFF@ #;
...
```

Figure 1: Marking locative information on the noun stem *-khaya*, which exhibits exceptional behaviour

In **ZulMorph** the noun stems are contained in `LEXICON NStem`. Xerox flag diacritics are employed to capture locative information as shown in the **lexc** script fragment in Figure 1. The flag diacritic `@P.LocSuf.OFF@` marks the noun stem *-khaya* as exhibiting idiosyncratic behaviour by not taking the locative suffix – see example 1(c) – while *-banga* is unmarked and will combine with the locative suffix *-ini*.

The procedure for obtaining locative information from the development corpus and capturing it in the MR lexicon entails the following:

- application of the morphological analyser to the corpus;
- automatic extraction of all analyses that contain either `[LocPre]` or `[LocPre] ... [LocSuf]`;
- human linguistic validation (see example 4);
- updating of the MR lexicon by including the validated locative information; and
- 'down translation' of the content in the MR lexicon for inclusion in the morphological analyser and the rebuilding of the morphological analyser.

This procedure is repeated until all words in the development corpus (and for that matter all available corpora) are correctly analysed and annotated with valid locative information. The analyses for *ebanga* 'while he was making' in 6(a) and *ebangeni* 'in the age-grade' in 6(b) are examples of overgeneration of the locative construction:

6(a) `ebanga`
`*e[LocPre]i[NPrePre5]li[BPre5] banga[NStem]`

6(b) `ebangeni`
`e[LocPre]i[NPrePre5]li[BPre5] banga[NStem]ini[LocSuf]`

6(c) `ebanga`
`e[SitSC1]bang[VRoot]a[VerbTerm]`

Whereas *ebangeni* in 6(b), which includes the locative suffix, is the correct locative form of the noun *ibanga*, and not *ebanga* as in 6(a), 6(c) reflects the correct analysis of *ebanga*. The implementation as shown in Figure 1 ensures the correct analyses.

Marking verbal extension information

In the verb root lexicon of the ultimate morphological analyser all verb roots should have their attested extensions indicated. It should be noted that in the case of the verbal extensions there is no clear known notion of *exception*, as is the case in the locative formation. So, instead of marking only exceptions, as before, *every root is marked with its attested extensions as found in corpora* or is marked to prevent known invalid extensions, as illustrated in example 3. This aspect has consequences for the scalability of any attempt at computationally associating every verb root with its own attested sequences of extensions.

In **ZulMorph** the verb roots are contained in `LEXICON VRoot`. As before, Xerox flag diacritics are employed to indicate known (sequences of) extensions to each verb root. This is shown in the **lexc** script fragment in Figure 2. The flag diacritics `@P.ExtAN.ON@`, `@P.ExtEK.ON@`, `@P.ExtIS.ON@`, `@P.ExtISAN.ON@`, etc., mark the verb root *-bon-* 'see' for its attested combination with the verbal extensions *-an-* (reciprocal), *-ek-* (neuter), *-is-* (causative) and *-isan-*, in turn appropriately marked with `@R.ExtAN.ON@`, `@R.ExtEK.ON@`, `@R.ExtIS.ON@`, `@R.ExtISAN.ON@`, etc. The flag diacritic `@P.ExtEK.OFF@` marks the verb root *-dum-* 'thunder' for not combining with the neuter extension *-ek-*.

Two aspects of Figure 2 may justify further clarification. Firstly, it shows a fragment of `LEXICON VExt`, as used in the extraction of idiosyncratic information regarding verbal extensions. One set of entries (extensions) caters for all unknown, yet-to-be-discovered valid combinations, and the other set caters for already attested or invalid marked combinations. These two sets of paths are clearly disjoint. By observing the resulting analyses it will be clear which new combinations have been discovered and warrant inclusion in the lexicon, since an attested extension will be marked with an `[M]` while a new one will occur with a `[U]`. Secondly, the cyclic nature of `LEXICON VExt` (it has itself as a continuation class) ensures that all combinations will be analysed and extracted. This forms a crucial part of the corpora mining process. Once all verb roots have been marked with their attested extensions, the annotations `[M]` and `[U]` will be removed.

```
! Multicharacter Symbols
[U] [M]
@P.ExtAN.ON@ @R.ExtAN.ON@ @D.ExtAN@
@P.ExtEK.ON@ @R.ExtEK.ON@ @D.ExtEK@
@P.ExtIS.ON@ @R.ExtIS.ON@ @D.ExtIS@
@P.ExtISAN.ON@ @R.ExtISAN.ON@ @D.ExtISAN@
@P.ExtAN.OFF@ @P.ExtEK.OFF@ @P.ExtIS.OFF@
...
LEXICON VExt
...
! As yet unknown extension combination paths
an[RecipExt][U]@D.ExtAN@ : an@D.ExtAN@ VExt;
ek[NeutExt][U]@D.ExtEK@ : ek@D.ExtEK@  VExt;
is[CausExt][U]@D.ExtIS@ : is@D.ExtIS@  VExt;
isan[CausExt][U]@D.ExtISAN@ : isan@D.ExtISAN@  VExt;
...
@R.Verb.ON@  VerbTerm;
...
! Attested marked extension combination paths
an[RecipExt][M]@R.ExtAN.ON@ : an@R.ExtAN.ON@ VExt;
ek[NeutExt][M]@R.ExtEK.ON@ : ek@R.ExtEK.ON@  VExt;
is[CausExt][M]@R.ExtIS.ON@ : is@R.ExtIS.ON@  VExt;
isan[CausExt][M]@R.ExtISAN.ON@ : isan@R.ExtISAN.ON@  VExt;
...
LEXICON VRoot
...
bon@P.ExtAN.ON@@P.ExtEK.ON@@P.ExtIS.ON@@P.ExtISAN.ON@ VPSClass15;
dum@P.ExtEK.OFF@@P.ExtEL.ON@ VPSClass15;
...
```

Figure 2: Marking extension information on the verb roots -bon- 'see' and -dum- 'thunder'

The procedure for extracting verbal extension information from the development corpus is, in principle, similar to the procedure followed for locative information.

Since there are no rules to determine which verb roots may combine with which (sequence(s) of) extensions (see Table 1), a verb root such as *-dum-* 'thunder' could be generated by **ZulMorph** with any of these, which are not necessarily all correct. An example is **-dum-ek-,* as given in example 3. This overgeneration can be limited by the implementation presented in Figure 2.

Results and discussion

The procedure was applied to the development corpus of 30 000 types. The application of the morphological analyser resulted in the identification of 347 noun stems in the corpus that form their locatives by the combination of prefixation and suffixation, while 167 noun stems were found to form their locatives by means of prefixation only. This information has subsequently been incorporated in the MR lexicon and morphological analyser. In the locative formation of nouns, regular as well as idiosyncratic construction information is attached to noun stems, which ensures that only correct forms are recognised or analysed.

Regarding verbal extensions, 1 055 out of a total of 1 361 verb roots in the corpus were found to occur with one or more verbal extensions. In the given corpus, sequences of up to four extensions were found to occur, as shown in Table 1.

Reiterating, the various verbal extensions are not compatible with all verb roots, and there are no hard and fast rules that determine the possible combinations (that is, roots with extensions, as well as extensions with one another). Corpus mining of combinations of extensions with verb roots therefore contributes to correct combinations and sequences being recognised or analysed. Such information is

Table 1: Examples of verbal extension sequences

is	ek	is	w	*-qin-is-ek-is-w-a*
an	is	el	w	*-ehluk-an-is-el-w-a*
el	is	w	an	*-val-el-is-w-an-a*
an	is	el	an	*-qath-an-is-el-an-a*

not available elsewhere – not even paper dictionaries provide complete information on combinations and sequences for all verb roots. Regarding some of the examples in Table 1, the latest monolingual Zulu dictionary (Mbatha, 2006: 1237) lists the verb root *-val-* 'close' with the extensions *-el-*, *-is-*, *-w-* (in this sequence) but not followed by *-an-* as a last extension in the sequence. Extension information for the verb root *-qath-* 'break up new soil/crunch hard' is even scarcer in the sense that no combinations or sequences of extensions are given. Instead, five extensions are merely listed, namely *-an-*, *-el-*, *-ek-*, *-is-*, *-w-* (Mbatha, 2006: 1023). Therefore combinations and sequences of extensions with verb roots mined from corpora have also been incorporated in the MR lexicon and morphological analyser.

Since the morphological analyser by design includes `LEXICONs` (lists) of noun and verb roots, it only analyses words based on roots that occur in this so-called underlying lexicon. The source from which these roots were extracted is based on a Zulu dictionary dating from the 1950s, therefore the coverage of words from a real corpus of running text can be disappointing if only these roots are available (also see Beesley, 2003: 33). What is therefore also needed is a regularly updated MR lexicon as a basic 'lexical database' from which all of the associated forms for the various entries can be derived. This is a valuable resource for the Zulu language, since it can be used as a foundation for high-quality morphological analysis, as well as a host of other higher level applications.

Conclusion and future work

This paper proposes a corpus-mining approach to containing overgeneration in Zulu finite-state computational morphology with respect to locative formation from nouns and verbal extensions to verb roots. A semi-automated procedure for extracting idiosyncratic information by means of a finite-state morphological analyser is discussed and illustrated by means of examples. A significant amount of such information was extracted from a chosen corpus and included in the MR lexicon and morphological analyser.

Future work entails the exploitation of a guesser variant (Beesley & Karttunen, 2003: 444–451) of the morphological analyser for the identification of further idiosyncrasies in regular morphophonological rules.

In the case of locatives, the phonological process known as consonantalisation normally takes place when the locative suffix is added to stems ending in *-o* or *-u*. Exceptions to this rule occur, however, and provision needs to be made for them, for example:

7(a) *e-in-ndlu-ini* > *endlini* 'in the house'
loc.pref-class9.noun.pref-noun.stem-loc suf
(not **endlwini* as expected)

There are also exceptions to the palatalisation rules, for example:

7(b) *e-isi-bhamu-ini* > *esibhamini* 'on the gun'
loc.pref-class7.noun.pref-noun.stem-loc.suf
(not **esibhamwini* as expected)

Regarding verbal extensions, rule-based palatalisation occurs in the formation of passives when the final syllable of a verb root begins with a bilabial consonant. However, idiosyncrasies occur when consonants appearing elsewhere in the verb root are palatalised, for example:

8 *-sebenz-w-a* > *-setshenzwa* 'being worked'
-verb.root-pass.ext-terminative
(not **-sebenzwa* as expected)

The necessity of the guesser variant becomes obvious in examples where the morphological analyser fails to analyse the word because the palatalisation rules for locative formation and verbal

extensions are violated. However, by means of human intervention the necessary information for these idiosyncrasies can be added to the MR lexicon.

In the longer term, the eventual use of **ZulMorph** in real-world NLP/HLT applications will largely be determined by its linguistic soundness and completeness, and by its suitability and usability as a software tool.

Notes

[1] A previous version of this article appeared in the *Proceedings of the 3rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistic 2007* (Vetulani, 2007).

Acknowledgement — This material is based upon work supported by the National Research Foundation (South Africa) under grant number 2053403. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Research Foundation.

References

Beesley KR. 2003. *Finite-state morphological analysis and generation for Aymara*. Proceedings of the Workshop on Finite-State Methods in Natural Language Processing, 10th Conference of the EACL03, 12 - 17 April 2003, Budapest, Hungary, pp 19–26.

Beesley KR & Karttunen L. 2003. *Finite state morphology*. Stanford, CA: CSLI Publications.

Bosch SE, Pretorius L & Jones J. 2006. *Towards machine-readable lexicons for South African Bantu languages*. Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006, 23 - 26 May, Genoa, Italy.

Cole R, Mariani J, Uszkoreit H, Zaenen A & Zue V. 1997. *Survey of the state of the art in human language technology*. New York: Cambridge University Press.

Hurskainen A. 1992. A two-level formalism for the analysis of Bantu morphology: an application to Swahili. *Nordic Journal of African Studies* **1**(1): 87–122.

Hurskainen A. 2004. Swahili Language manager: a storehouse for developing multiple computational applications. *Nordic Journal of African Studies* **13**(3): 363–397.

Mbatha MO (ed.). 2006. *Isichazamazwi SesiZulu*. Pietermartizburg: New Dawn Publishers.

Poulos G & Msimang CT. 1998. *A linguistic analysis of Zulu*. Pretoria: Via Afrika.

Pretorius L & Bosch SE. 2003a. Finite-state computational morphology: an analyzer prototype for Zulu. *Machine Translation – Special issue on finite-state language resources and language processing* **18**: 195–216.

Pretorius L & Bosch SE. 2003b. Computational aids for Zulu natural language processing. *Southern African Linguistics and Applied Language Studies – Special issue on language technology in Southern Africa: resources and applications* **21**(4): 267–282.

Van Eeden BIC. 1956. *Zoeloe-Grammatika*. Stellenbosch: Universiteitsuitgewers en Boekhandelaars.

Vetulani Z (ed.). 2007. *Proceedings of the 3rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistic*, 5 - 7 October 2007, Poznan, Poland.